(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
16 January 2003 (16.01.2003)

PCT

(10) International Publication Number
WO 03/005192 A1

(51) International Patent Classification[7]: G06F 9/445, 9/46, 11/20, 15/177

(21) International Application Number: PCT/SE02/01353

(22) International Filing Date: 4 July 2002 (04.07.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
0102405-8 4 July 2001 (04.07.2001) SE

(71) Applicant *(for all designated States except US)*: **IDÉKAPITAL AB** [SE/SE]; Box 5718, S-114 87 Stockholm (SE).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **REIMER, Markus** [SE/SE]; Ejdergatan 6, S-619 32 Trosa (SE). **OSSIANSSON, Magnus** [SE/SE]; Fridhemsvägen 1-3, Lästringe, S-610 60 Tystberga (SE).

(74) Agents: **ALBIHNS STOCKHOLM AB** et al.; Box 5581, Linnégatan 2, S-114 85 Stockholm (SE).

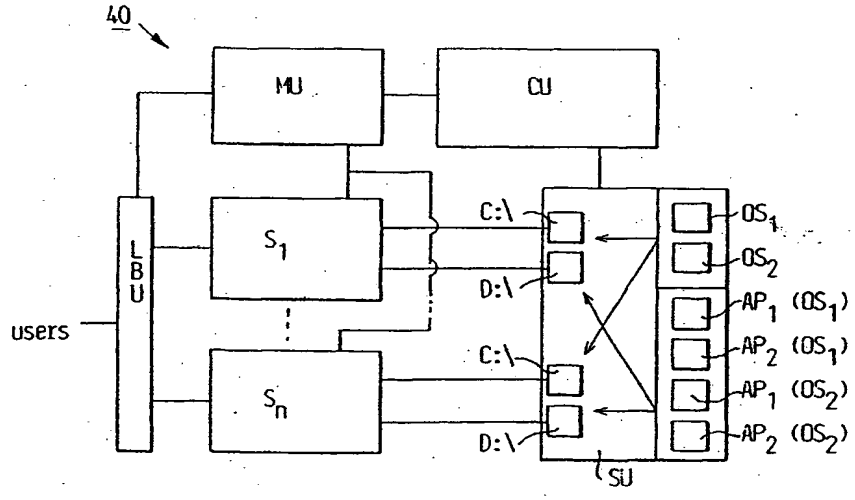
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: A SYSTEM AND A METHOD FOR SELECTING A PRECONFIGURED OPERATING SYSTEM FOR A SERVER



(57) Abstract: In a pool of servers, the capacity may be used in a better way if the servers can be rebooted with different operating systems and provided with different sets of appli-cation depending on the current needs in the network. It may be decided to allocate more machine capacity to an application, remove machine capacity from an applica-tion or move capacity from one application to another. The servers can be allocated dynamically to a particular combination of operating system and applications. Therefore, a service provider can guarantee access to an application at all times without keeping excess capacity at times when the load on the application is low. Also, the service provider does not have to provide a spare server for each combi-nation of operating system and application in case of failure. One spare server can be used to replace different servers depending on the needs at any given time.

WO 03/005192 A1

A SYSTEM AND A METHOD FOR SELECTING A PRECONFIGURED OPERATING SYSTEM FOR A SERVER

Technical Field

The present invention relates to an arrangement for use in a computer network for providing at least one service to at least one client computer, as defined in the preamble of claim 1. The invention also relates to a method for use in a computer network for providing at least one service to at least one client computer as defined in the preamble of claim 7 and a computer program product as defined in the preamble of claim 13.

Background and Prior Art

A service provider usually provides different services, such as software applications, to customers using several types of operating systems, such as Windows, Unix, Linux, etc. A customer using a particular operating system cannot normally use a service provided by a server using another operating system. The service provider must therefore provide several servers with different operating systems to provide services to different customers. Supporting only one type of operating system would be less expensive, but would not enable providing services to computers using other operating systems.

The load in a system varies with time. For example, during the night the load is lower than during the day. In most cases the load has a peak of considerably higher load than the rest of the day, during a relatively short time span. The load pattern varies in dependence of the application and also in dependence of other factors. To ensure access to the system at all times, including the peak load, the service provider must ensure a capacity that is higher than what is needed most of the time.

Usually, the service provider also has a spare system in case one of the servers fails and must be repaired. At least one spare server must be available for each supported operating system. This is an expensive solution for securing up-time of the system.

All these tree factors cause the need for keeping excess capacity; that is, an excessive number of servers are idle most of the time.

Another major concern is the security issues. A user accessing the server can perform unauthorized actions that may cause damage to the server and thereby affect the function of the server, which in turn will affect other users currently accessing the same server. This problem does not only apply to service providers as such, but could apply to any entity managing a network having at least one server that is accessed by many users.

Summary of the Invention

It is therefore an object of the present invention to minimize the need for hardware installations while ensuring the capacity needed to meet the requirements on a system at any given time.

This object is achieved according to the invention by an arrangement for use in a computer network for providing at least one service to at least one client computer, said client computer using a first operating system, said arrangement comprising at least a first (S1) and a second server (S2) accessible by the client computer, each of said servers being connectable to a first memory location (C:\), said arrangement comprising:

storage means comprising at least a first (OS1) and a second (OS2) preconfigured operating system stored in such a way that it can be retreived by the at least one server, but cannot be altered by an unauthorized user,

control means for monitoring the function of the at least first and second server,

control means for initiating a reboot of the first server,

control means for selecting an operating system with which to reboot the first server,

control means for downloading the booting information of the selected preconfigured operating system to the first memory location of the first server and booting the server using the downloaded booting information

The object is also achieved according to the invention by a method for use in a computer network for providing at least one service to at least one client computer, said client computer using a first operating system, said arrangement comprising at least a first (S1) and a second server (S2) accessible by the client computer, each one of said servers being connectable to a first memory location (C:\), each one of said first and second server being connectable to storage means comprising at least a first (OS1) and a second (OS2) preconfigured operating system stored in such a way that it can be retreived by the at least one server, but cannot be altered by an unauthorized user, monitoring the function of the at least first and second server,
determining if said first server needs to be rebooted,
if the first server needs to be rebooted, selecting an operating system with which to reboot the first server,
downloading the booting information of the selected preconfigured operating system to the first memory location of the first server and
booting the server using the downloaded booting information

The object is also achieved according to the invention by a computer program product for use in a computer network for providing at least one service to at least one client computer, said client computer using a first operating system, said arrangement comprising at least a first (S1) and a second server (S2) accessible by the client computer, each of said servers being connectable to a first memory location (C:\), said first and second server being connectable to first storage means comprising at least a first (OS1) and a second (OS2) preconfigured operating system stored in such a way that it can be retrieved by the at least one server, said computer program product comprising code means which, when it is executed in a computer, will make the computer perform the following functions:

determining if said first server needs to be rebooted,
if the first server needs to be rebooted, selecting an operating system with which to reboot the first server,
downloading the booting information of the selected operating system to the first memory location of the first server and
booting the server using the downloaded booting information

With the inventive arrangement, method and program the preconfigured operating system for a server can easily be restored or changed depending on the need of the users. The number of servers running a particular operating system can be easily adapted to the need at any given time.

Preferably the arrangement further comprises storage means comprising at least a first and a second version of a first application, said first version being adapted to the first operating system (OS1) and said second version being adapted to the second operating system (OS2),
control means for selecting the version of said at least one application adapted to the selected preconfigured operating system to be downloaded to the first server, said application being
control means for downloading and installing the at least one application to the first memory location of the first server .

In this embodiment the method further comprises the steps of
selecting the version of said at least one application adapted to the selected preconfigured operating system to be downloaded to the first server,
downloading and installing the at least one application to the first memory location of the first server

Further, the code means of the computer program product is arranged to make the computer perform the following steps:

selecting the version of said at least one application adapted to the selected preconfigured operating system to be downloaded to the first server,
downloading and installing the at least one application to the first memory location of the first server.

In this embodiment, one or more applications may be provided to the server after rebooting the server, so that the server can provide a particular set of applications to users accessing the network. This makes the system more flexible regarding the capacity for providing different applications. Since the peak load may occur at different times depending on the type of application, several applications can share a pool of servers. The peak load times also vary geographically, because of the time zones. The excess capacity can be allocated to the application that currently has a peak. When the load changes, the excess capacity can be allocated to another application currently experiencing a peak.

In a preferred embodiment the arrangement also comprises a load balance unit (LBU) connected to each of the servers, for directing new users to one of the servers.

The decision to reboot one or more servers may be taken in dependence of the load in the network. The operating system and, if applicable, application or applications, may be selected in dependence on the load in at least one of the servers.

A service provider typically signs a service level agreement with a customer, i.e. a content provider, where the service provider offers to ensure access through the Internet to the content provider's services. With the arrangement and method according to the invention, the servers can be allocated dynamically to a particular combination of operating system and applications. Therefore, a service provider can guarantee access to an application at all times without keeping excess capacity at times when the load on this particular application is low. Also, the service provider

does not have to provide a spare server for each combination of operating system and application in case of failure. Instead, the same spare server can be used to replace different servers depending on what is needed at any given time. In other words, fewer spare servers are needed than the supported number of operating systems, which in turn will reduce the cost for providing redundancy functionality for the system.

Another advantage is that the servers may be rebooted on a regular basis with an uncorrupted preconfigured operating system to minimize the danger of malfunction in the servers caused by user violation etc.

In this document the term "machine" means either a hardware unit having certain properties, such as an operating system and support functions for an application, or a virtual machine existing on one or more hardware units. Further, several virtual machines can share the same hardware.

The invention therefore takes advantage of the fact that different applications have different requirements for machine resources. According to the invention three different types of decision may be made:

Decisions to allocate more machine capacity to an application

Decisions to remove machine capacity from an application

Decisions to move capacity from one application to another

These decisions may be based on one or more of the following criteria, or other criteria:

- The number of users of a particular application. If there are many users, or the number of users is increasing, the hardware resources allocated to the application may have to be increased. If there are few users, or the number of users is decreasing, it may be possible to reduce the hardware resources allocated to the application.

- The response time for the application. If the response times for an application are long, it may be necessary to increase the machine capacity allocated to the application.
- The number of machines currently allocated to the application. A maximum or minimum number may be specified in the level of service agreement, or may be made dependent on the load.
- The geographical location of the application
- The logical location of the application
- The load on a particular machine. If the load is high, the possibility to allocate more resources to the applications running on this machine should be investigated. If the load is low, it should be considered whether to set part of the machine resources allocated to the application in standby mode instead, or to allocate the machine resource to another application.
- Security alarm on a physical or logical machine. In this case the machine may have to be taken down, which means that it should be replaced by another machine if necessary.
- Time in operation. A reboot can be performed automatically after a certain uptime.
- Planned events, such as backup or maintenance, may call for a reallocation of applications to another machine.
- An operator may decide to allocate machine resources to an application, to remove machine resources from an application or to move machine resources from one application to another.
- An indication of an increase or decrease in the need for resources.
- An indication of excess capacity for a particular application may lead to a reduction in the hardware machines allocated to the application. The hardware removed from the application may be put in standby mode or allocated to another application.

- The time for releasing a machine may be taken into account. If it would take a very long time to make a machine ready to be taken down, it will in most cases be better so choose another machine.
- The time for starting a new application may also be taken into account when deciding whether to start the application or not.

Some or all of these criteria may be specified in a service level agreement, as mentioned above. For example, the service provider may guarantee a maximum response time at any given time. The service level agreement can also specify a maximum capacity that is to be guaranteed regardless of the load.

The result of any of these decisions can be a request for the allocation of a new machine resource to an application. It can also be a request to remove a hardware resource with or without allocating it to another application. Allocation of resources can be denied, for example because the maximum amount of resources specified in the service level agreement has already been allocated.
For any of these decisions a message may be sent to the customer.

Brief Description of the drawings

Fig. 1 shows a first embodiment of a system according to the present invention.
Fig. 2 shows a second embodiment of a system according to the present invention.
Fig. 3 shows a third embodiment of a system including two servers.
Fig. 4 shows a fourth embodiment of a system including an undefined number of servers.
Fig. 5 shows a flow chart of a method for rebooting a server according to an embodiment of the invention.
Fig. 6 shows a flow chart of a method for rebooting a server, including moving users according to an embodiment of the invention.
Fig. 7 shows a flow chart of the third embodiment of the present invention.

Detailed description of preferred embodiments

Figure 1 shows a first embodiment of a system 10, including a server S1, having a first memory location C:\, the C drive, which may be a separate hard drive or a partition of a hard drive. The server is connected to a control unit CU, and to a storage unit SU. The storage unit may contain memory circuits, hard drives etc. The purpose of the storage unit is to store selected information that must be accessible to the system 10. The control unit CU is also connected to the storage unit 20 SU. The first memory location C:\ may also be located in the storage unit SU, as shown in fig. 2.

A plurality of users may access the server S1 and store user information, such as documents, presentations, etc, at a predetermined location H:\ (H drive) in the system. The user information may be stored in the server S1 (not shown) in a separate or as a partition of a hard drive, but is preferably stored outside the server, e.g. in the storage unit SU, as indicated by H:\ in Fig. 1, or in a separate file server (see fig. 2). Each user will only see the drives that the server 51 has mapped up, in this example C and H drive.

The storage unit SU includes a second memory location comprising booting data of a plurality of preconfigured operating systems OS1, OS2, in this example two standard operating systems, each being preconfigured with a specific type of applications, such as word processing applications. Examples of standard operating systems are MS Windows NT, Linux, Unix etc. The second memory location may also be implemented in the server S1 as one or several separate hard drives (not shown).

The users will not have access to the second memory location, to avoid any accidental or intentional damage to the server S1.

Figure 2 shows a second embodiment 20 of the present invention including a server S1, a control unit CU, a storage unit SU, a separate file server FS and a monitoring unit MU. The first memory location C:\ is, in this embodiment located in the storage

unit SU together with the preconfigured operating system OS1, OS2 in the second memory location, The control unit CU is connected to the storage unit SU and the monitoring unit MU. The server S1 is connected to the monitoring unit MU, the storage unit SU and the file server FS.

The monitoring unit helps the control unit CU to monitor the number of users accessing the server, monitor the load in the server, measures the up-time of the server, monitor any accidental or voluntary unauthorised action in the server by monitoring data consistency and/or detecting server data intrusion, etc. The control unit may also control the server S1 via the monitoring unit MU.

Figure 3 shows a third embodiment 30 of the present invention comprising two servers S1 and S2. Both servers are connected to a load balance unit LBU, which controls the users access to each server, as described in more detail below. Each server is also connected to a storage unit SU, where the first memory location C:\ of each server is located. The storage unit SU has, in this example, booting data for three different preconfigured operating systems OS1, OS2, OS3 for each server S1, S2.

As described in connection with figure 2, the system is also provided with a monitoring unit MU and a control unit CU, connected to the storage unit SU and the monitoring unit MU. The monitoring unit MU is also connected to each server and to the load balance unit LBU. The function of the load balance unit is well known in the field, and it assists with directing new users to an appropriate server. The access to a server for new users may easily be controlled by the load balance unit. A new user will be directed to a particular server offering the desired service with the appropriate operating system. If more than one server fulfils the requirements, the load on each of these servers will be considered when directing the user.

The embodiment 30 is also provided with means to store user information, such as a file server or similar, but since this is not a part of the invention it is not shown for the sake of clarity.

Figure 4 shows a fourth embodiment of the present invention, comprising a large number of servers (S1-Sn), n=6, n is larger than 1, all connected to the load balance unit LBU, as described in connection with figure 3, and to the storage unit SU. The function of the control unit CU and the monitoring unit MU is as described above, but the storage unit SU in this embodiment comprises a third memory location storing information regarding different applications for each standard operating system, In this example there are two different types of application configurations for each operating system: AP1(OS1), AP1 (OSl), AP2 (OS2), and AP2 (OS2). Examples of application configurations may be Word processing applications, database applications, Economy applications, etc. Alternatively, for service providers on the internet, the applications may be e-commerce applications, games or other programs, or applications providing information, such as news services.

In figure 4 only two servers are shown, but the dotted line between them indicate that there may be more server, and in this example n is assumed to be 6. Servers S1-S4 are the primary servers where the servers S1 and S2 have a first standard operating system OSl, e.g. MS Windows NT, and the servers S3 and S4 have a second standard operating system OS2, e.g. Linux. Servers S1 and S3 are preconfigured with applications APl of a first type and servers S2 and S4 are preconfigured with applications AP2 of a second type. This way the system provides support for two different standard operating systems, each having two application configurations. The load balance unit only directs the users to the appropriate server meeting the need of each user.

Servers S5 and S6 are maintained as spare servers, in case one of the primary servers S1-S4 breaks down or if there is an unexpected peak load for a specific applica-

tion in one standard operating system, e.g. AP1(OS1) word processing applications using Linux. If the load on the server S1 becomes too high, one of the spare servers S5 may be configured to provide the same service. In this case, the spare server S5 is booted using OS1 and AP1, as will be discussed in detail in connection with Fig. 7.

A flow diagram, shown in figure 5, describes the method for selecting a preconfigured operating system in the embodiments shown in Figures 1 and 2, that is, when selecting only an operating system.

The process starts in step 50

Step S1: The control unit CU selects a preconfigured operating system (OS1 or OS2). This selection may be determined in several ways, as will be discussed below.

Step 52: The next step of the process to reboot the server is to copy the booting data of the selected preconfigured operating system from the secure second memory location to the first memory location C:\.

Step 53: The server boots up by using the uncorrupted data that was copied to the first memory location C:\.

The flow ends in step 54.

The most obvious use of the process shown in Figure 5 is to reboot the server S1 regularly by measuring the up-time of the server. When a first predetermined time has passed, preferably less than 1 hour, e.g. 10-15 minutes (but the predetermined time may be extended to be up to for instance 100 hours or more), a flag is set in the control unit CU, indicating that the server should be rebooted as soon as possible. This is preferably performed when no users are accessing the server S1. If there are users accessing the server all the time, and the up-time of the server passes a second predetermined time, e.g. twice as long as the first predetermined time, a second flag may be set in the control unit CU, indicating that a request should be sent to the present users to log out and that no new users may be logged on to the server, The re-

booting will take place as soon as the users have logged out or after a predetermined time, e.g. 10 minutes, after the request is sent, thereafter throwing out any users that have not logged out yet.

If the control unit detects an accidental or voluntary unauthorised action in the server, e.g. someone tries to get unauthorised access to the server the server is shut down immediately and the users are thrown out, to minimise the damage the intruder may achieve. The control unit thereafter selects the same preconfigured operating system as before to reboot the server using uncorrupted booting data. The server may also be rebooted for other reasons, for example, to change the operating system.

If a privileged user has access to the server, the user may request a change of preconfigured operating system, thereby causing the control unit CU to select the desired preconfigured operating system and rebooting the server after the user has logged out from the server.

Figure 6 is a flow chart describing the process of the invention, applied to the embodiment shown in Figure 3.

The process starts in step 60 and proceeds to step 61.

Step 61: The servers S1 and S2 are monitored by the monitoring unit MU, measuring up-time, number of users, load, unauthorised use etc. of each server, as described in connection with figures 1 and 5.

Step 62: The control unit CU uses the information obtained in step 61 to determine if a server needs to be rebooted and which preconfigured operating system should be used.

The decision regarding whether any server should be rebooted is taken in this step, which will be discussed in more detail below. If there is no server to reboot, the process returns back to step 61, where the monitoring

unit continues to monitor the servers. If the control unit CU makes a decision to reboot a server the process continues to step 63.

Step 63: The control unit removes present users from the selected server by performing the following:

1) send an order to the load balance unit, via the monitoring unit, to direct new users to an alternative server having the same operating system as the server being shut down. If no alternative server is available, the control unit may boot a spare server (not shown), by using the process described in connection with figure 5, to create a server with the same preconfigured operating system as the one that should be shut down and rebooted. If there is no spare server available, new users will be denied access to any server until the server is rebooted.

2) Wait until present users, accessing the server, are logged out or, if more urgent, send a request to the present users to log out as soon as possible or, if extremely urgent, throw out present users. This is indicated in box 63a and 63b, where a decision is made in box 63a to proceed to box 63c if no users are logged on to the server or if an urgent need to throw out present users has occurred (priority changed). On the other hand if there are still users present on the server and the priority is not changed, a notification is sent to the users requesting that they should log off, see box 63b. An alternative solution is to hand over present users and their running applications to an alternative server (if available),

3) Shut down the server.

When the server is shut down in box 63c, the process continues to step 64,

Step 64: A preconfigured operating system is selected, as will be discussed in more detail below.

Step 65: The booting data needed to reboot is copied from the uncorrupted second memory location to the first memory Location C:\ of the server.

Step 66: The server is booted using the booting data stored in the first memory location C:\. The process returns back to step 61 after the server has been booted up.

Steps 62 and 64 are essential for determining if a server should be rebooted and which operating system should be installed during the rebooting process. These steps are preferably implemented in a rule matrix, where the input parameters may be up-time, number of users, load and unauthorised use and/or others, as discussed above. By combining the decision of the rule matrix with the change of load on the input to the servers appropriate measures may be taken. The taken measures always depend on the result from the monitoring process of the servers.

A server should be rebooted with the same preconfigured operating system if, for instance, the up-time has exceeded a predetermined time, as described earlier, or if an unauthorised use has been detected.

A server should be rebooted with another preconfigured operating system if, for instance, there is a need for more capacity in another preconfigured operating system. A such example could be when the number of users and/or the load of a server exceed a predetermined level. If there is no spare server available, the server having a low number of users is selected to be rebooted with another preconfigured operating system to increase the capacity. The present users of the server being rebooted, may be transferred to another server, if available, or thrown out if the need is urgent.

Users will be denied access to the server if there is an escalating trend in number of users etc., which indicate that the capacity in the other preconfigured operating system will not be sufficient in a near future, and the server is rebooted to increase the capacity.

Figure 7 is a flow chart of the method according to the invention, applied to the embodiment shown in Figure 4. The following will happen in case of a failure in one of the primary servers, e.g. server S3.

The process starts at step 70 and proceeds to step 71.

Step 71: The monitor unit MU monitors all 20 servers (S1-Sn) as described earlier. When server S3 fails the monitor unit registers this and alerts the control unit CU.

Step 72: A decision to boot up one of the spare servers, e.g. S5, is taken, since the server S3 may be experiencing a hardware failure.

Step 73: Any present users are removed from the server S3. This may be done rapidly if needed or during a longer period of time as described in connection with Fig. 6 and the server is shut down in box 73a.

Step 74: A standard operating system is selected, in this example OS2,

Step 75: The booting data of the selected standard operating system is copied from the second memory location to the first memory location of server S5. In this example there is only provided a second memory location for all the servers. This saves memory space and makes it easier to update the different operating systems.

Step 76: An application configuration, in this example AP1(OS2), is selected to replace the server S3.

Step 77: This information regarding the selected application configuration is thereafter copied to an additional memory location D:\.

Step 78: The server is thereafter booted using the booting data from the C drive and the application configuration on the D drive. The process returns back to step 71.

When the spare server is up and running, any new users requiring the second type of operating system for the first type of applications, i.e. AP1(OS2), are directed to server S5 by the load balance unit, which is controlled by the control unit CU via the monitoring unit MU. No new users are directed to the failing server S3.

As may be seen from this example there is no need to provide a separate spare server for each server having a specific operating system and application configura-

tion. Only one spare server is needed, but preferably half as many spare servers as primary servers should be provided to provide a higher quality-of-service level.

Another advantage with this system is that it reduces the energy consumption, since fewer spare servers are needed, and the servers are used in a more cost efficient and environmentally friendly manner, since a spare server may substitute a number of different servers.

Figure 7 may also be used when illustrating what happens during an unexpected peak load, that is, when the capacity for a particular application should be increased.

Step 71: The monitoring unit indicates that a specific server, e.g. server S2, experiences a number of users exceeding a predetermined level, e.g. 400 users.

Step 72: If there are any spare servers available in idle mode, the control unit CU decides to boot up (at least) one spare server. In this example only S6 is available and idle, since S5 has previously taken over the workload of server S3, which has failed.

Step 73: If necessary, users are removed from the spare server. In this case, no users have to be removed from the spare server S6 and the server does not need to be shut down in step 73a.

Step 74: If the control unit CU has not already selected the operating system, in this example OS1, this is done here.

Step 75: Then the booting data is copied from the second memory location to the first memory location of server S6.

Step 76: If the application configuration has not already been selected, in this example AP2, this is done here.

Step 77: The application configuration is copied to the additional memory location D:\ in step 77.

Step 78: The spare server S6 is thereafter booted up using the booting data from the C drive and the application configuration on the D drive.

The process returns back to step 71.

If there are no available idle spare servers, any other server being idle, i.e. having no users, may be reconfigured by first shutting down the server and rebooting the server with the desired operating system and application configuration.

A flexible system has thus been achieved where it is possible to change the operating system of any server, reboot a server that has a long up-time on a regular basis, meet peak load, reduce the number of back-up servers and increase the security of a server system.

The described invention may be implemented as a server sharing system for a service provider having a large number of servers to fulfil the need and requirements of the clients, but the invention should not be limited to this. It is fully possible to implement the invention in any network where there is a need to regularly reboot the server(s) and sometimes even to change the operating system of a server, such as a company network.

The invention also has the advantage that the life span of a server is increased due to the cycling of the servers (rebooting due to long up-time) and it also increases the security of the system since an uncorrupted version of the preconfigured operating system is downloaded each time the server is rebooted.

In this description the word "users" is intended to cover both persons and machines using the resources and services provided by the servers. The methods described above are preferably implemented as a software application that generally includes instructions to carry out the method steps described in connection with figures 5, 6 and 7. The software application is preferably stored in the control unit CU during operation, but may naturally be provided on a CD-ROM, diskettes or other memory storage media, which the control unit may access for running the application.

In all the embodiments discussed here, the function of the control unit, the monitoring unit and the load balance unit is controlled by computer programs run on the appropriate unit.

Claims

1. An arrangement for use in a computer network for providing at least one service to at least one client computer, said client computer using a first operating system, said arrangement comprising at least a first (S1) and a second server (S2) accessible by the client computer, each of said servers being connectable to a first memory location (C:\), said arrangement comprising:

storage means comprising at least a first (OS1) and a second (OS2) preconfigured operating system stored in such a way that it can be retreived by the at least one server, but cannot be altered by an unauthorized user,

control means for monitoring the function of the at least first and second server,

control means for initiating a reboot of the first server,

control means for selecting an operating system with which to reboot the first server,

control means for downloading the booting information of the selected preconfigured operating system to the first memory location of the first server and booting the server using the downloaded booting information

2. An arrangement according to claim 1, further comprising

storage means comprising at least a first and a second version of a first application, said first version being adapted to the first operating system (OS1) and said second version being adapted to the second operating system (OS2),

control means for selecting the version of said at least one application adapted to the selected preconfigured operating system to be downloaded to the first server, said application being

control means for downloading and installing the at least one application to the first memory location of the first server .

3. An arrangement according to claim 1 or 2, further comprising a load balance unit (LBU) connected to each of the servers, for directing new users to one of the servers.

4. An arrangement according to any one of the preceding claims, further comprising control means arranged to initiate a reboot of said first server in dependence of the load in the network.

5. An arrangement according to any one of the preceding claims further comprising control means arranged to select an operating system in dependence on the load in at least one of the servers.

6. An arrangement according to any one of the claims 2-5, further comprising control means arranged to select the version of the application in dependence on the load on at least one application in the server.

7. A method for use in a computer network for providing at least one service to at least one client computer, said client computer using a first operating system, said arrangement comprising at least a first (S1) and a second server (S2) accessible by the client computer, each one of said servers being connectable to a first memory location (C:\), each one of said first and second server being connectable to storage means comprising at least a first (OS1) and a second (OS2) preconfigured operating system stored in such a way that it can be retreived by the at least one server, but cannot be altered by an unauthorized user, monitoring the function of the at least first and second server,
determining if said first server needs to be rebooted,
if the first server needs to be rebooted, selecting an operating system with which to reboot the first server,
downloading the booting information of the selected preconfigured operating system to the first memory location of the first server and

booting the server using the downloaded booting information

8. A method according to claim 7, wherein said servers are connectable to storage means comprising at least a first and a second version of a first application, said first version being adapted to the first operating system (OS1) and said second version being adapted to the second operating system (OS2),
said method further comprising the steps of
selecting the version of said at least one application adapted to the selected preconfigured operating system to be downloaded to the first server,
downloading and installing the at least one application to the first memory location of the first server

9. A method according to claim 7 or 8, further comprising the step of determining the need for reboot in dependence on the load in the network.

10. A method according to any one of the claims 7-9, further comprising the step of selecting the operating system with which to reboot the first server in dependence of the load on at least one of the operating systems.

11. A method according to any one of the claims 8-10, further comprising the step of selecting the version of the application in dependence of the load on the at least one application.

12. A method according to any one of the claims 7-11, further comprising the step of directing a user accessing the network to one of said servers by means of a load balancing unit (LBU).

13. A computer program product for use in a computer network for providing at least one service to at least one client computer, said client computer using a first operating system, said arrangement comprising at least a first (S1) and a second

server (S2) accessible by the client computer, each of said servers being connectable to a first memory location (C:\), said first and second server being connectable to first storage means comprising at least a first (OS1) and a second (OS2) preconfigured operating system stored in such a way that it can be retrieved by the at least one server, said computer program product comprising code means which, when it is executed in a computer, will make the computer perform the following functions:

determining if said first server needs to be rebooted,

if the first server needs to be rebooted, selecting an operating system with which to reboot the first server,

downloading the booting information of the selected operating system to the first memory location of the first server and

booting the server using the downloaded booting information

14. A computer program product according to claim 13, wherein said network further comprises storage means comprising at least a first and a second version of a first application, said first version being adapted to the first operating system (OS1) and said second version being adapted to the second operating system (OS2), said code means being arranged to make the computer perform the following steps:

selecting the version of said at least one application adapted to the selected preconfigured operating system to be downloaded to the first server,

downloading and installing the at least one application to the first memory location of the first server

15. A computer program product according to claim 13 or 14 wherein the code means will initiate a reboot of the first server in dependence of the load in the network.

16. A computer program product according to any one of claims 13-15, wherein the operating system will be selected in dependence of the load on at least one operating system.

5

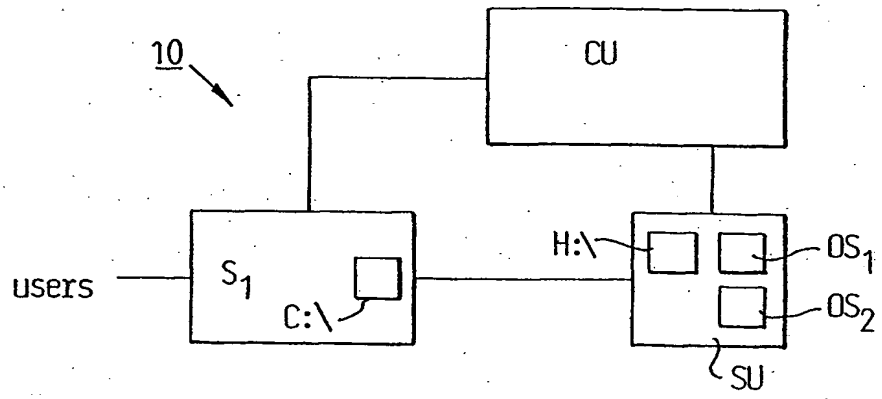


FIG.1

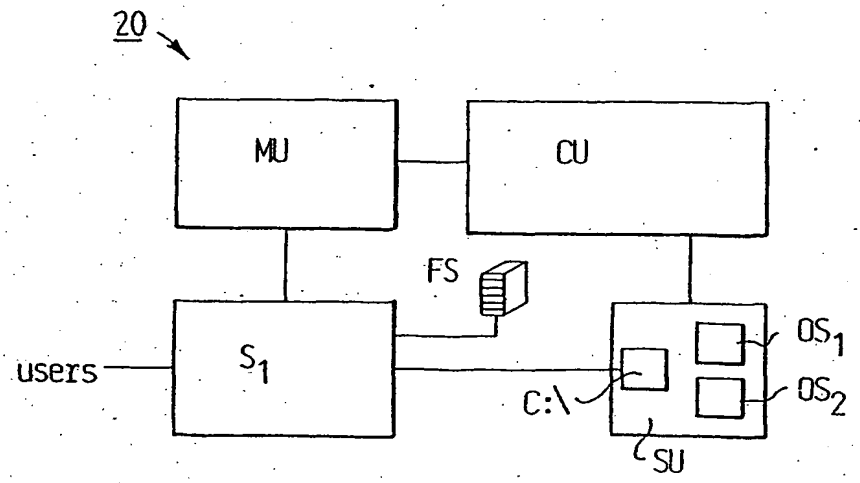


FIG.2

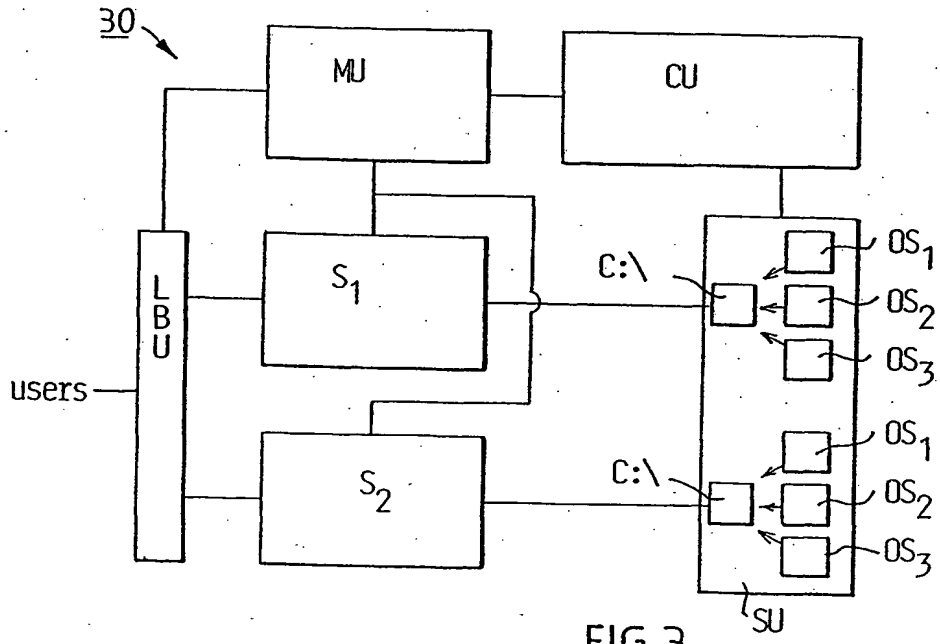


FIG. 3

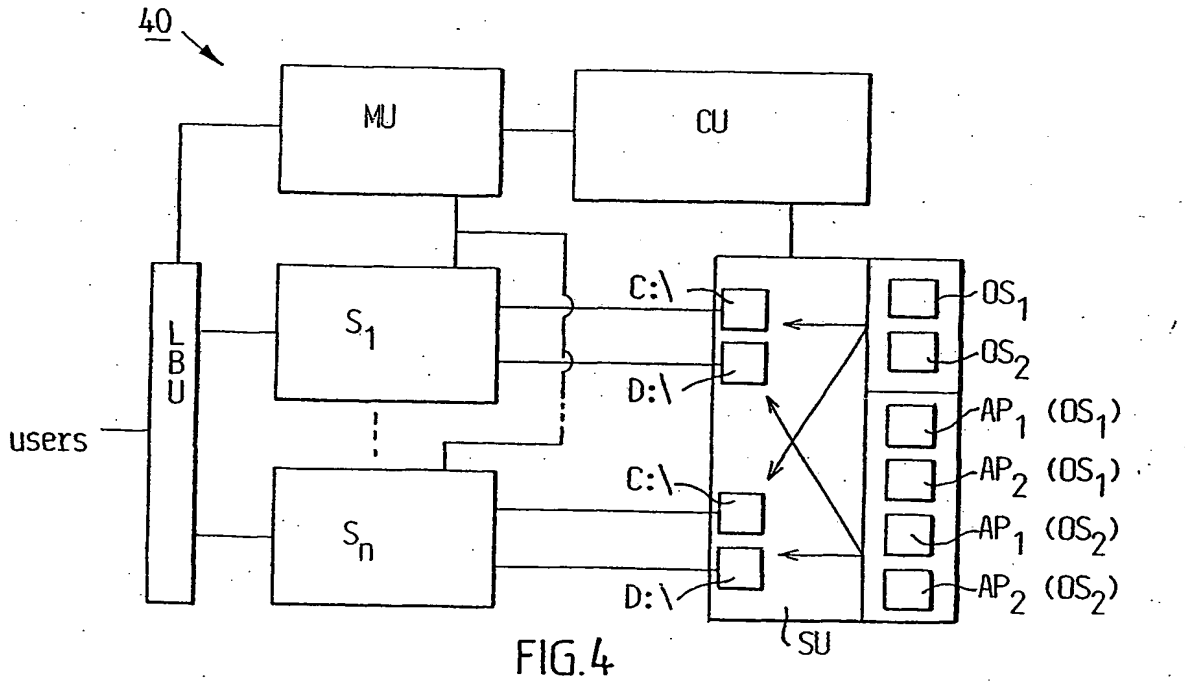


FIG. 4

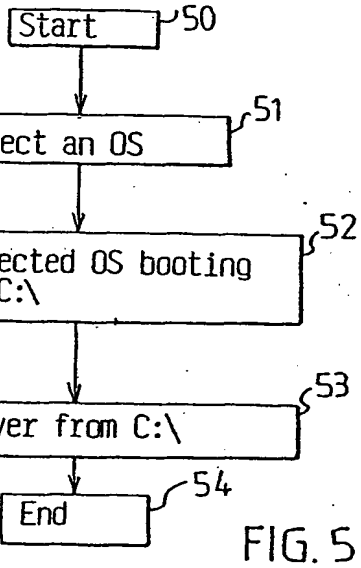

Start
50
Select an OS
51
Copy selected OS booting data to C:\
52
Boot server from C:\
53
End
54

FIG. 5

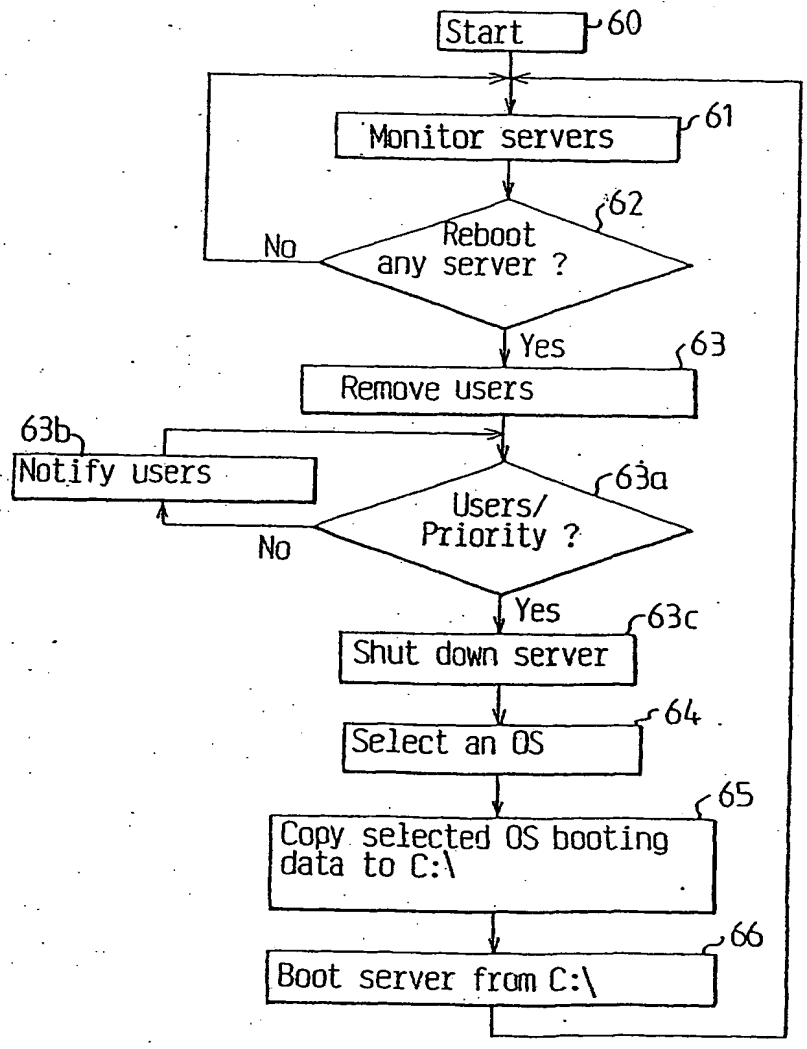

Start
60
Monitor servers
61
Reboot any server ?
62
No
Yes
Remove users
63
63b
Notify users
Users/ Priority ?
63a
No
Yes
Shut down server
63c
Select an OS
64
Copy selected OS booting data to C:\
65
Boot server from C:\
66

FIG. 6

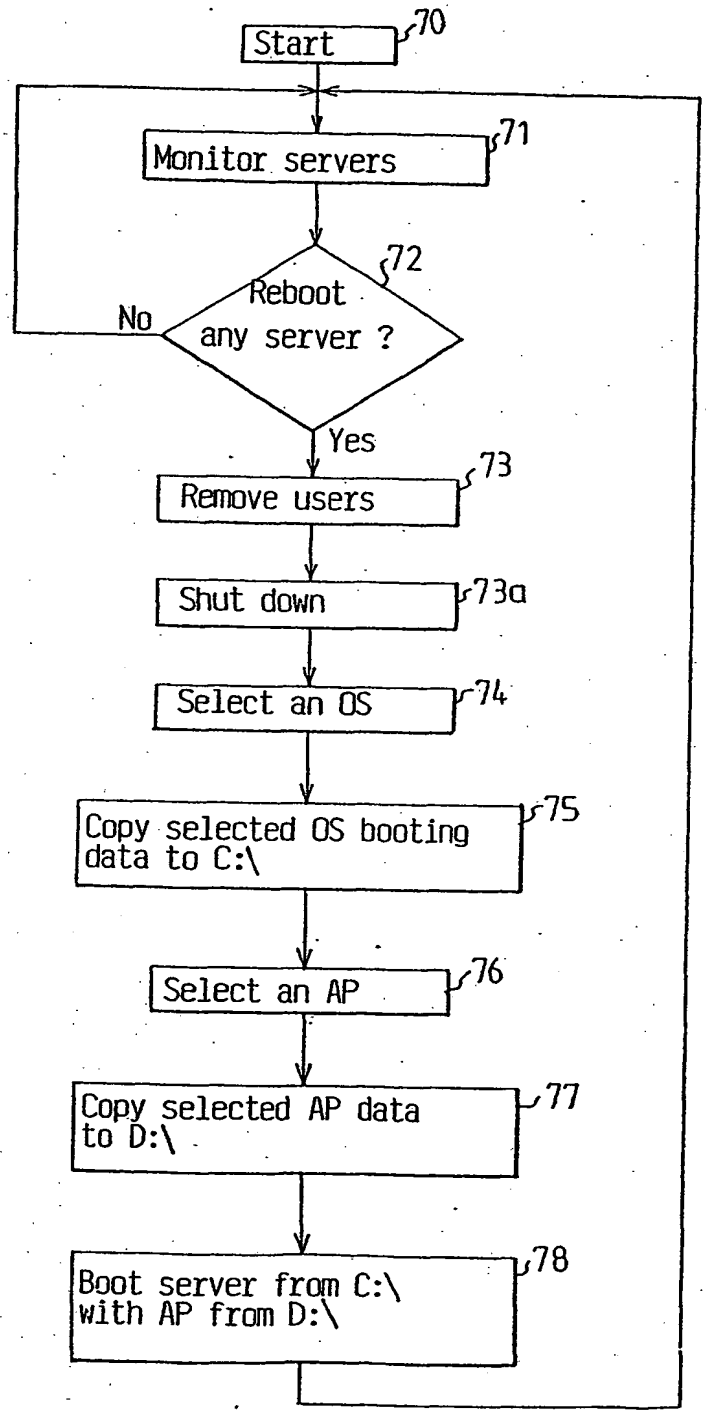


FIG. 7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/SE 02/01353

A. CLASSIFICATION OF SUBJECT MATTER

IPC7: G06F 9/445, G06F 9/46, G06F 11/20, G06F 15/177
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC7: G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

SE,DK,FI,NO classes as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-INTERNAL, WPI DATA, PAJ, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5675723 A (EKROT, A.C. ET AL.), 7 October 1997 (07.10.97), column 2, line 15 - line 56; column 5, line 19 - line 47; column 12, line 55 - column 13, line 24, figure 3	1-3,7-8, 12-14
Y	EP 0838753 A1 (SUN MICROSYSTEMS, INC), 29 April 1998 (29.04.98), column 9, line 35 - line 56	1-3,7-8, 12-14
A	US 5742829 A (DAVIS, M.L. ET AL.), 21 April 1998 (21.04.98), abstract	2,6,8,11,14

[X] Further documents are listed in the continuation of Box C. [X] See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
7 October 2002	10-10-2002
Name and mailing address of the ISA/ Swedish Patent Office Box 5055, S-102 42 STOCKHOLM Facsimile No. +46 8 666 02 86	Authorized officer Jenny Forss/LR Telephone No. +46 8 782 25 00

Form PCT/ISA/210 (second sheet) (July 1998)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/SE 02/01353

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 6134673 A (CHRABASZCZ, M.), 17 October 2000 (17.10.00), abstract ---	2,6,8,11,14
A	GB 2346715 A (CHEN-CHANG SU), 16 August 2000 (16.08.00), page 3, line 7 - page 4, line 20 ---	1,7,13
A	EP 1037133 A1 (INTERNATIONAL BUSINESS MACHINES CORP), 20 Sept 2000 (20.09.00), column 4, line 53, abstract --- --------	1,7,13